This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims**

1.      (Currently Amended) A method for monitoring the performance of applications running on a plurality of servers in a distributed computing environment, comprising:

        providing a user with at least two choices as to a level of monitoring,

        receiving from a user selected information for monitoring, wherein the selected information for monitoring includes user selection of one of a level of monitoring and particular features to be monitored, wherein each level of monitoring provides a different level of detail, wherein the selected information for a first level of monitoring comprises request level data and server level data, wherein the selected information for a second level of monitoring includes the selected information for the first level and API level data, wherein the selected information for a third level of monitoring includes the selected information for the second level of monitoring and method level data, wherein the user is provided with an option of changing dynamically between the second level and the third level,

        receiving from the user identification of a schedule for monitoring of the selected information, wherein the schedule consists of a group of schedule records, each of which is a combination of a start date and time and a monitoring level, wherein the schedule define times for a monitoring level to change,

        monitoring application  performance in accordance with the selected information and in accordance with the identified schedule, wherein the monitoring commences with a first schedule record, wherein the monitoring changes when a current time is a start time and date of another schedule record, and wherein the monitoring continues through successive schedule records, and

        making monitored performance information available to the user in accordance with the selected information.


2.      (Original) The method of claim 1, wherein the user is prompted to identify a scope of information to be monitored, and wherein application server performance is monitored in accordance with the selected scope.

3. (Currently Amended) The method of claim 2, wherein said scope comprises a first monitoring level ~~wherein the selected information comprises request level data and server level data~~, wherein the selected information includes availability management including information as to whether a particular application is running on a particular server, system resources including information as to an amount of available memory and a number of available connections, and basic request data including a number of requests being made and a number of requests being completed.

4. (Currently Amended) The method of claim 3, wherein said scope further comprises a second monitoring level wherein the selected information further comprises <u>the</u> API level data for problem determination for servers with a high volume of transactions and occasional instability<u>, wherein the API level data includes data regarding throughput of a particular CPU and functionality to permit the user to provide a soft cancel of a request</u>.

5. (Currently Amended) The method of claim 4, wherein said scope further comprises a third monitoring level, wherein the selected information further comprises <u>the</u> method level data for problem determination for servers that have been selected for diagnostics, detailed workload characterization and profiling.

6. (Cancelled)

7. (Currently Amended) A method for monitoring the performance of applications running on a plurality of servers in a distributed computing environment, comprising:

prompting a user to select a <u>type of trap</u> ~~server or server group, a resource, and a threshold or condition for notification~~,

in response to the prompting, receiving from the user information used for one of a threshold condition type of trap, a number of hits type of trap, and a resource consumption type of trap, wherein a trap is capable of providing a notification or alert to the user, <u>wherein the information includes a resource and a condition in a form of a threshold value for the threshold condition type of trap, wherein the information includes a resource and a condition for the</u>

number of hits type of software trap, wherein the information includes an application server or
server group, a resource, and a threshold for the resource consumption type of trap,

comparing the value or quality of a parameter to the threshold or condition, and

[[if]] in response to the parameter ~~reaches~~ reaching the threshold or condition,

logging information concerning the parameter,

determining whether an alert condition has been triggered, and

in response to determining that the alert condition has been triggered,

communicating the alert condition to the user.


8.      (Previously Presented) The method of claim 7, further comprising comparing the
parameter to criteria for notifying a user, and notifying a user if the criteria are met.


9.      (Currently Amended) The method of claim 7, ~~wherein the threshold or condition is a
value of a resource,~~ wherein the threshold value depends on the particular type of resource ~~and
further comprising:~~

~~in response to the threshold or condition being met, evaluating whether an alert condition
has been triggered, and~~

~~in response to the alert condition having been triggered, communicating the alert
condition to the user~~.


10.     (Original) The method of claim 9, wherein the resource is a property of a method.


11.     (Original) The method of claim 10, wherein the resource is CPU time.


12.     (Currently Amended) The method of claim 7, wherein the threshold or condition is a
number of hits, and further comprising:

in response to a request or statement meeting the threshold or condition, incrementing a
hit counter, and

in response to sufficient hits being counted, reaching [[an]] the alert condition.

13.    (Original) The method of claim 12, wherein the resource is a request, and the condition is a string contained in the request.


14.    (Original) The method of claim 7, wherein the resource relates to application server performance, and the condition is percentage of CPU time.


15.    (Currently Amended) A method for monitoring the performance of applications running on a plurality of servers in a distributed computer system, comprising:

    providing the user with performance information,

    receiving from a user a selection of one of request analysis, method analysis, SQL analysis, and application server analysis,

    in response to the user selection of the request analysis, the method analysis, or SQL analysis,

        prompting the user to select a metric, wherein the metric is based on the user selection of one of for the request analysis [[,]] and for the method analysis includes one of throughput, response time, and CPU time, wherein the metric for the SQL analysis includes one of throughput and response time, and wherein the metric for the application server analysis includes one of pool size, concurrent waiters, average wait time, faults, percentage pool usage, physical connections, free memory, and memory used, and

        in response to receiving the metric from the user, providing a trend analysis to the user,

        receiving from the user a request for more specific performance information, and providing more detailed performance information.


16.    (Currently Amended) The method of claim 15, further comprising receiving from the user selection of server availability analysis and, in response to the user selection of the server availability analysis, providing a new trend analysis to the user.


17.    (Cancelled)

18.     (Currently Amended) The method of claim 16, further comprising receiving a request for a decomposition report on a portion of the trend analysis, and providing a decomposition report, wherein the decomposition report for the report analysis and the method analysis represents breakdown of a trend report from the trend analysis by request type or request name, or by server name if the trend report applies to a server group, wherein the decomposition report for the SQL analysis report represents breakdown of the trend report by request name, request type, method name, or table name, wherein the decomposition report for the server availability analysis represents breakdown of the trend report for a group by application server.

19.     (Currently Amended) The method of claim 18, further comprising receiving a request for a detail report on the portion of the decomposition report, and providing a detail report from the decomposition report for the request analysis, the method analysis, or the SQL analysis.

20-35. (Cancelled)

36.     (Previously Presented) The method of claim 1, further comprising:
        providing, in a system running at least one application, a management application having various components for monitoring and management, and
        monitoring and providing to a user in real-time information concerning configuration of the components and the relationships between the components.

37-41. (Cancelled)

42.     (Cancelled)

43-50. (Cancelled)

51-52. (Cancelled)

53.     (Cancelled)

54-55. (Cancelled)

56-71. (Cancelled)

72.     (Cancelled)

73-75. (Cancelled)

76-79. (Cancelled)

80.     (Cancelled)